WEST

| Help | Logout | Interrupt |

| Main Menu | Search Form | Posting Counts | Show S Numbers | Edit S Numbers | Preferences | Cases |

Search Results -

Terms	Documents
L18 and 17	28

Database:

```
US Patents Full-Text Database
US Pre-Grant Publication Full-Text Database
JPO Abstracts Database
EPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins
```

Search:

```
L19
```

| Refine Search |

| Recall Text ⬍ | | Clear |

Search History

DATE: **Wednesday, February 06, 2002**   Printable Copy   Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=TDBD; PLUR=YES; OP=OR		
L19	L18 and l7	28	L19
L18	L16 and l4	40	L18
L17	L16 and (l5 or l6)	2	L17
L16	(view$ or read$ or look$ or scan$ or brows$ or examin$ or perus$ or scrutini$ or inspect$ or display$) near5 l8	119	L16
L15	l2 and l3 and l4 and (l5 or l6)	7	L15
L14	l2 and l4 and (l5 or l6) and l10 and l11	1	L14
L13	L12 and l3	2	L13
L12	l2 and l4 and (l5 or l6 or l7) and l10 and l11	34	L12
L11	break$ or pagebreak$ or databreak$	2883	L11
L10	tuple$ or record$ or row$ or bank$	11074	L10
L9	policy or policies or role$	805	L9
L8	(part$ or section$ or portion$ or piece or pieces or element$ or fragment$ or segment$ or page or pages) near5 l1	1319	L8
L7	authoriz$ or authoris$ or permission$ or permit$ or clear$ or approv$ or allow$ or right$ or privilege$	36062	L7
L6	security adj (clearance or authori$) or permissions	122	L6
L5	(access or clearance or security or authori$ or confidenti$ or permission or usage or privilege$) near3 (level$ or grade or grades or status or standing or rank$ or degree$ or score$ or rating or class$ or categor$ or credential$ or right$)	513	L5
L4	tag or tags or identifier$ or mark$ or marking$ or label$ or symbol$	13459	L4
L3	(view$ or read$ or look$ or scan$ or brows$ or examin$ or perus$ or scrutini$ or inspect$ or display$) near5 l1	1294	L3
L2	(creat$ or produc$ or generat$ or construct$ or mak$ or develop$ or form$ or forming or formation or build$ or craft$ or assembl$ or structur$ or format$ or put$ adj together) near5 l1	7662	L2
L1	(report$ or document$ or page$ or webpage$)	17929	L1

END OF SEARCH HISTORY

WEST

☐ | Generate Collection | | Print |

L19: Entry 1 of 28   File: TDBD   Jan 1, 1997

TDB-ACC-NO: NN9701179

DISCLOSURE TITLE: Publicly Accessible Web Pages with Restricted Direct Link Access

PUBLICATION-DATA:
IBM Technical Disclosure Bulletin, January 1997, US

VOLUME NUMBER: 40
ISSUE NUMBER: 1
PAGE NUMBER: 179 - 180

PUBLICATION-DATE: January 1, 1997 (19970101)

CROSS REFERENCE: 0018-8689-40-1-179

DISCLOSURE TEXT:

Disclosed are methods to create pages on the World Wide Web which are publicly accessible but include portions that are not directly accessible. A visitor to the web site can thus be restricted to viewing the pages in certain orders. In particular, the methods eliminate the potential for viewing a subpage in unknown context, e.g., via a link from another web site or via a hotlist entry. The first method utilizes multiple directories. One or more directories contain files to which outside links are to be permitted; for files in the remaining directories, no outside links are intended. Frequently, for example, every night a program scans through the no-outside-links directories and renames each file to some random set of characters. The program then scans each file in each directory, finding hypertext links and references to the former name of each file, and updating all such links and references to point to the new file name. The random file names are chosen to avoid duplication. The result is that all links from other sites and hotlist entries work if pointing to files in the first set of directories, but the only working links to files in the second set of directories will be those links from within the directories updated by the program. Pseudocode for the program used by the first method follows, showing the simple case where only two directories are involved. Directory LINKME includes all files intended to be linkable from anywhere on the web. Directory NOLINKME contains the files which are to be visible anywhere on the web but whose names will be periodically changed to prevent their inclusion in hotlists and external links. NEWNAMES = "" OLDNAMES = "" For each file in NOLINKME Add the file name to the list OLDNAMES end NEWNAME = randomly chosen name For each file name in OLDNAMES Do while NEWNAME is included in NEWNAMES or OLDNAMES

NEWNAME = randomly chosen name end Append NEWNAME to the list NEWNAMES end For each file in NOLINKME and each file in LINKME For I=1 to the number of file names in OLDNAMES Replace all links pointing to the Ith file name in OLDNAMES with a link pointing to the Ith file name in NEWNAMES end end For I=1 to the number of file names in OLDNAMES

Rename file with the Ith name in OLDNAMES to the Ith name in NEWNAMES end A second method involves the use of a Computer Generated Imagery (CGI) script for each link that is intended to be permanently accessible from anywhere on the web. When a request arrives, the CGI script generates the file appropriate for that particular link. In addition, the CGI script checks to see if files exist for the local links within the file just served. If the files do not exist, the CGI script creates them and starts a countdown timer. If the files already exist, the CGI script resets the countdown timer. In either case, when the timer pops the files are deleted. The countdown period for the timer would be chosen to allow a reasonable time interval for reading the first file and requesting all the other files; typically, this would

be on the order of minutes or hours. A third method involves the use of a "REQ=" tag to indicate that the browser is required to be displaying a particular page before the desired page can be loaded. In this implementation a new tag, "REQ=", is used in an Hypertext Markup Language (HTML) file to specify which Uniform Resource Locator(s) (URL(s)) are required before a page can be accessed. After requesting a page, the browser checks the currently displayed page's URL against the REQ= tag(s) in the requested page's HTML. If a match is found, the requested page is loaded. If no match is found but one or more REQ= tags exist, the browser will pop up a page informing the user that a page indicated by the REQ= tag(s) must be accessed before the requested page can be seen. On this page will also be links to the required pages. A fourth method is essentially the same as the third method except that in this case the server does the checking on the REQ= tag. In this case, the server checks the browser's current location by reading the REFERER_URL variable and compares it to the URLs specified by any REQ= tags in the requested HTML. If the REFERER_URL matches a required URL, the page is served normally. If the server doesn't find a match and one or more REQ= tags exist in the requested HTML, the server will present a page informing the user that another page must be accessed first and will give links to the required page or pages.